Multimed Tools Appl (2016) 75:10617–10630
DOI 10.1007/s11042-014-2114-z

Two-stage neural network regression of eye location in face images

Krzysztof Rusek · Piotr Guzik

Received: 29 December 2013 / Revised: 9 May 2014 / Accepted: 16 May 2014/
Published online: 12 July 2014

Abstract Automatic eye localization is a crucial part of many computer vision algorithms for processing face images. Some of the existing algorithms can be very accurate, albeit at the cost of computational complexity. In this paper, a new solution to the problem of automatic eye localization is proposed. Eye localization is posed as a nonlinear regression problem solved by two feed-forward multilayer perceptrons (MLP) working in a cascade. The input feature vector of the first network is constructed from coefficients of a two dimensional discrete cosine transform(DCT) of a face image. The second network generates corrections based on small image patches. Feature extraction and neural network prediction have known and efficient implementations, thus the entire procedure can be very fast. The paper hints at the neural network structure and the procedure for generating artificial training samples from a low number of face images. In terms of accuracy, the method is comparable to state-of-the-art techniques; however it is based on numerical procedures that could be highly optimized (fast Fourier transform and matrix multiplication).

Keywords Eye localization · Neural network · DCT · Computer vision

1 Introduction

The problem of automatic eye localization first arose several years ago, along with the invention of automatic facial recognition algorithms. In fact it was the facial recognition research that motivated the development of eye localization algorithms. It was quickly realized that the accuracy of most facial recognition algorithms strongly depends on the correct

K. Rusek · P. Guzik (✉)
Department of Telecommunication, AGH University of Science and Technology,
Mickiewicza 30, 30-059, Kraków, Poland
e-mail: guzik@kt.agh.edu.pl
URL: http://www.kt.agh.edu.pl

K. Rusek
e-mail: krusek@agh.edu.pl

alignment of the detected face. This problem was broadly investigated in [12]. Sometimes satisfactory results can be obtained without face alignment [14]. Having said that, it is reasonable to assume that further alignment can only improve accuracy. The face alignment using the position of the eyes seems to be a natural choice here.

Although the automatic eye localization problem has been studied for a few decades, its performance still needs improvement. The main issue with this task is the variability of eye appearance; in an image, this not only depends on individual differences, but also on the lighting conditions, pose, expression, etc.

In [13] we present another approach to the problem of eye localization, where the issue is described a one of regression. DCT coefficients were proposed as a feature vector, and an artificial neural network was used for regression. This paper summarises previous results and extends them with the following: larger and much variable face collection, deeper neural networks, and two-stage regression - a mechanism substantially improving accuracy.

The rest of the paper is organized as follows. Section 2 describes related work on eye localization. In Section 3, our solution is presented. Results are discussed in Section 4. Finally, Section 5 concludes the paper.

2 Related work

Automatic localization of the eyes (or any part of the face) in a face image can be treated as a classification or regression task. In the classification approach, a classifier is trained to recognize whether a given part of an image contains an eye. This classifier is then applied to the sliding window at different scales. The combined results from neighbouring windows and scales define the final rectangle containing the eye. Any classifier can be used, but since the algorithm is repeated many times, its speed is an important factor.

The most common approach is to use the AdaBoost algorithm with Haar or LBP features [18, 19]. This exact technique is used in the popular computer vision library, OpenCV, in an implementation of the eye detection algorithm. All other algorithms that use template matching also fall into this category.

While the classification approach is much more common, the problem was also stated as a regression one [5]. In this formulation the eye locations are estimated from the feature vector $\boldsymbol{x}$ extracted from the face image.

Everingham and Zisserman limited their analysis to the kernel regression model, with the subset of pixels being the features. In this paper, we use a similar approach; however, instead of using a subset of the image we reduced the dimensionality by taking the most important DCT coefficients as the feature vector. Having considered performance, accuracy and model simplicity, we decided to use an artificial neural network for regression. A similar approach was presented in [9] where the authors suggest using MLP with DCT from YCbCr planes for classification.

Recently, Sun et. all [16] independently of [13] proposed a similar idea of using convolutional neural networks for facial point detection. The location of the points is given by the first network (regression problem) and it is corrected by another two convolutional networks. This solution is very powerful and general (five points are obtained at once); however, convolutional networks are known to be more difficult to train compared to multi-layer perceptrons. Our solution is simpler and faster in both the training and evaluation steps. A quite similar approach was studied in [15]. However, we experimented with DCT features and deeper networks that allowed us to obtain significantly better accuracy. It may be possible to improve the accuracy in a similar way as in [16].

3 Proposed approach

Following Everingham and Zisserman [5] we formulate the problem of eye location as a multinomial regression problem. Using a feature vector $\boldsymbol{x}$ we want to predict the output vector $\boldsymbol{y}$ representing the coordinates of both eyes. This is the standard regression problem, and a variety of solutions have been proposed to deal with it. They include artificial neural networks and support vector regression (SVR), state-of-the-art nonlinear regression algorithms. Since SVRs are commonly defined just for one dimensional predicted variable, they require four independent models, one for each component. As such, we used neural networks. In this case, the single model gives multidimensional prediction. Furthermore it is significantly more compact, and faster to evaluate, than a support vector machine with the same generalization performance [2]. The usefulness of neural networks for such tasks is presented in [10], where the authors use neural networks to refine eye localization indicated by a different algorithm.

3.1 Feature selection

When it comes to image processing, pixel intensity is the obvious candidate as a feature. However, it is highly dimensional, making the model complicated and difficult to train. It is possible to reduce the dimensionality by resizing the image, although in this a case some useful information about the face is lost. The other problem with resizing the image is that the estimated eye coordinates have to be scaled to the coordinates of the original image. Due to this scaling, the regression error is also scaled and grows linearly with the scale.

All these problems vanish if the right features are selected. Coefficients of the two dimensional discrete cosine transform(DCT) B_{pq} of a square image A_{mn} of size N given by Eq. (1) appear to have all the properties required to construct the feature vector.

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A_{mn} \cos\left(\frac{\pi(2m+1)p}{2N}\right) \cos\left(\frac{\pi(2n+1)q}{2N}\right) \tag{1}$$

$$\alpha_p = \begin{cases} \sqrt{\frac{1}{N}} & p = 0 \\ \sqrt{\frac{2}{N}} & p \neq 0 \end{cases} \qquad \alpha_q = \begin{cases} \sqrt{\frac{1}{N}} & q = 0 \\ \sqrt{\frac{2}{N}} & q \neq 0 \end{cases}$$

They contain all the information stored in the original image, although in the DCT domain it is much easier to remove unimportant details while shape information is preserved.

Eye location is determined by the orientation and shape of the head. Since we are processing images coming from the face detector, the largest object in the image is the head. Thus only a small number of low frequency DCT coefficients are needed to contain all the information about head location and orientation (see Fig. 1). This means it is possible to reduce the dimensionality of the feature vector.

Fig. 1 64×64 Face image before (*left*) and after(*right*) removing small DCT coefficients. Only 210 coefficients remain nonzero

Another property of DCT coefficients is that they are not invariant under rotation or translation. Something that may present a problem for some tasks is expected in eye location task since faces are often translated and rotated. DCT coefficients are not just affected by the geometrical transformation; different lighting conditions result in substantially different coefficients of the DCT transform. To make the algorithm resistant to such changes, some processing is required. In the proposed algorithm we applied image histogram equalization and the results we obtained proved to be quite satisfactory.

3.2 Neural network regression

For the regression, we used a standard feed-forward multilayer perceptron. In the first approach, we used two hidden layers. Later on, the size of the network was increased. The second hidden layer increases accuracy without noticeable memory consumption during the training phase compared to a larger network with a single hidden layer. The architecture is presented in Fig. 2. Later on, for clarity, the network configuration will be described numerically e.g. configuration 18-18 for the network in Fig. 2. Each unit in the hidden layer has a tansig activation function

$$tansig(n) = \frac{2}{1 + e^{-2n}} - 1, \tag{2}$$

while the output layer is purely linear.

In order to make neural network training more efficient, a pre-processing was performed on inputs and targets [8]. Inputs (DCT coefficients) were normalized to have zero mean and unit variance, while the outputs (eye coordinates) were normalized in such a way that the lower left corner of an image had coordinates $\left(\frac{1}{N}, \frac{1}{N}\right)$, where N is the image size(it is not (0, 0) because in Matlab indexes start from 1). The top right corner is at (1, 1).

Neural networks just like any other supervised learning algorithms require labelled training examples. The bigger the training set, the more complex the network that can be fitted. However, care must be taken with designing very complex models, since neural networks are prone to overfitting in such a case.

Fortunately there is an easy way of multiplying the number of training examples by a factor of a few hundred. For each image in the database and the associated eye coordinates, we can apply affine transforms to get the new training examples.

Not all of the transforms are valid. Some of them may introduce such a deformation to the face that the resulting image would be unrealistic. Therefore we permitted only *translation*

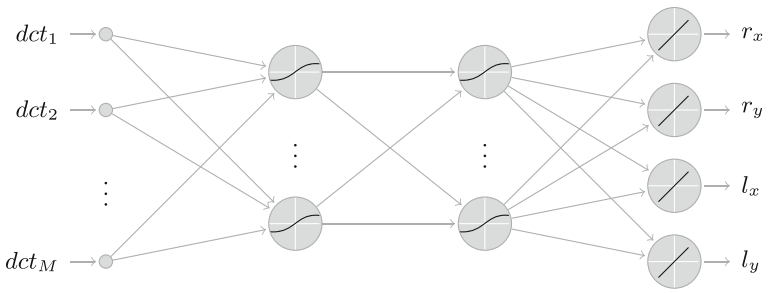

Fig. 2 Neural network structure. Hidden layers have a tansig activation function and the output layer is purely linear

by a vector (b_1, b_2), *rotation* by an angle θ and *scaling* by a scale parameter a. All permited transforms of a two dimensional vector x can be expressed in matrix notation as

$$x' = a \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} x + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \quad (3)$$

By varying parameters a, b_1, b_2 and θ, we can generate as many examples as needed from a single labelled image. The parameters have to be chosen carefully to generate plausible images i.e. such as one would expect at the output from the face detector.

Using these artificial training examples it is possible to train a larger network with the higher generalization power compared to the network trained on the original database. However, the examples generated by this procedure cannot span the entire feature space since no out of plane rotations are involved. This is why large labelled training datasets like [1] are still needed.

3.3 Two-stage regression

In our further work we were using more diverse dataset. In fact, we prepared this dataset with use of two publicly availabla databases. Complete description of the construction of this dataset referred to as *Set2* is presented in in Section 4.1. We noticed that a simple tri-layer network presented in Fig. 2 was not sufficient to achieve good performance on Set2.

We arrived at an alternative solution. Instead of using more features, we introduced a second stage neural network, fed by pixels. The first stage network is fed by DCT coefficients and finds a coarse-grained eye localization. The second stage network is fed by the $\pm 7px$ image patch around the location indicated by the first network. Thus the problem of high dimensionality of pixel data is removed, while its accuracy is retained. In the second stage, we used a network of configuration 20-20-20. The size of the network was reduced to compensate for the larger feature vector. The above mentioned choice of feature vector ($15 \times 15 px$ image patch) means that the network has 225 inputs. The configuration 20-20-20 requires 5402 network parameters.

4 Numerical experiments

The previous section describes the problem and the concept of the proposed solution. This section contains more details about the model, its parameters and training procedure. Here we also discuss quantitative results.

4.1 Model training

Eye localization using neural network regression and DCT coefficients can be realized in many different ways. In particular, it is necessary to select the number M of DCT coefficients, number of hidden layers in the network and the number of neurons in each layer.

The more coefficients or neurons are used, the higher the accuracy of the model can be obtained. However, a large number of parameters requires a large number of training samples to avoid overfitting. This can be problematic even when the previously described training sample generation procedure is applied. The second problem is that a large training set requires a large amount of memory (frequently running into tens of GB) during the training phase.

With all these restrictions in mind, we need to find parameters that result in a fairly accurate model with a reasonable training set size. The first parameter to tune is the number of DCT coefficients. We decided to use the complete upper left triangle of the image in the dct domain. We noticed that when the triangle size (corresponding to the number of selected coefficients) was too small, an individual was unable to correctly identify the eyes in the reconstructed image. We found that the triangle size 15×15px is the smallest that is sufficient to correctly localize eyes in an image. This indicates 210 dct coefficients as features. This is clearly shown in Fig. 1. This number of features is used as an input to the neural network.

Having selected the feature dimensionality, it is possible to experiment with different topologies of the neural network. In our previous work, we found that a fully connected 2 hidden layer network with 18 neurons in each layer gives satisfactory results. However, such a solution is sub-optimal, and it should be possible to find an alternative that will achieve better accuracy. We found that the network with two hidden layers has better accuracy compared to the network with a single layer, and a comparable number of neurons. Moreover, less memory is required for training such a network than a network with a single layer and 36 neurons. Thus we end up with the network shown in Fig. 2. Here we have extended the network to four hidden layers with 20 neurons in each layer. The previous network proved insufficient for our new experiments where, we merged two significantly different face image databases. This deeper network has a configuration 20-20-20-20.

As mentioned previously, there is a need for an appropriate number of training examples. The rule of thumb is to have at least approx ten times more training samples than the number of parameters in the considered model. The network from Fig. 2 has 4216 parameters, thus 40,000 training required. We decided to increase this number to $\sim 100,000$ samples since 30 % of them are used as test and validation sets during the neural network training phase. It should also be noted that adding another layer to the neural network does not result in a significant increase in the number of parameters in the considered model. For the configuration 20-20-20-20, the network has 5564 parameters, and $\sim 100,000$ training samples are sufficient.

A typical database of face images in which the eye positions are labeled contains approc a thousand images. Therefore, from each image in the database, a hundred examples need to be generated. We picked a hundred random transformations of type (3) for every training image. The parameters of the transformations were uniformly distributed in the following ranges $a \in (0.8, 1.2)$, $\theta \in \left(-\frac{\pi}{16}, \frac{\pi}{16}\right)$, $b_1, b_2 \in (-15, 15)$.

The networks were trained using the neural network toolbox in MATLAB. We used the Levenberg-Marquardt algorithm with a mean squared error (mse) of eye coordinates as a performance function. The charts of error rates are presented in Figs. 3, 4, 5. Note, that mse cannot be directly related to normalized eye localization error, which is the main accuracy measure described later in the paper.

4.2 Computational aspects

The main advantage of the proposed eye localization method is its speed. The implementation of neural network relays on matrix multiplication and DCT can be efficiently calculated using the fast Fourier transform. Both operations are extremely fast and optimized on many platforms.

Let us compare our approach to a typical sliding window algorithm. For face images of size 64×64 px the eye is within a range 15×15 to 25×25 pixels. Assuming a typical window growth factor of 1.1, it is required to perform 6 stages of search (15,17,19,20,22,25).

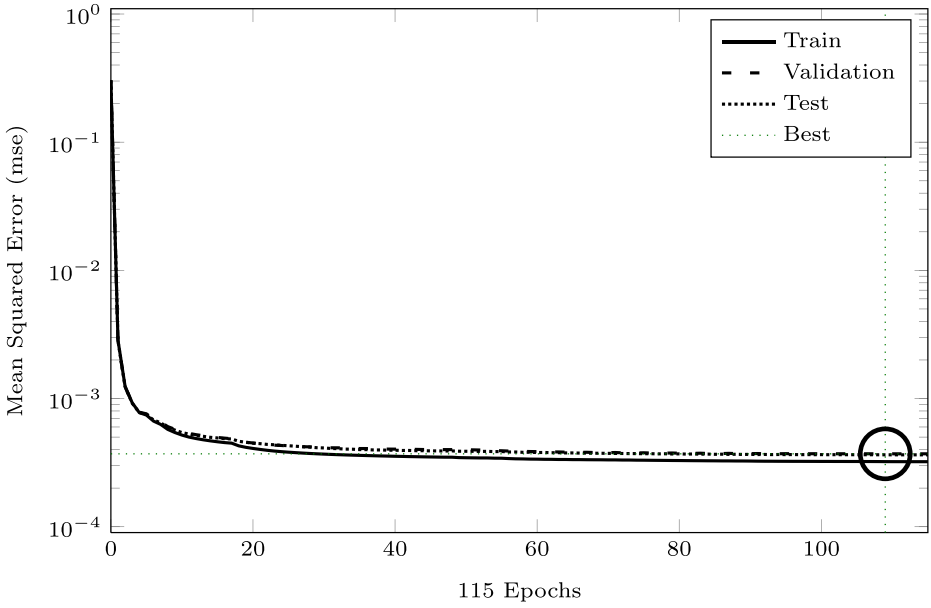

Fig. 3 Performance for stage 1. Best Validation Performance is 0.00037035 at epoch 109

If the window is shifted by approx. 20 % of its width w in each stage, there are $\sim \left[\left(\frac{64}{w} - 1\right)/0.2\right]^2$ windows to classify. This results in approx. 900 classification operations for the entire image.

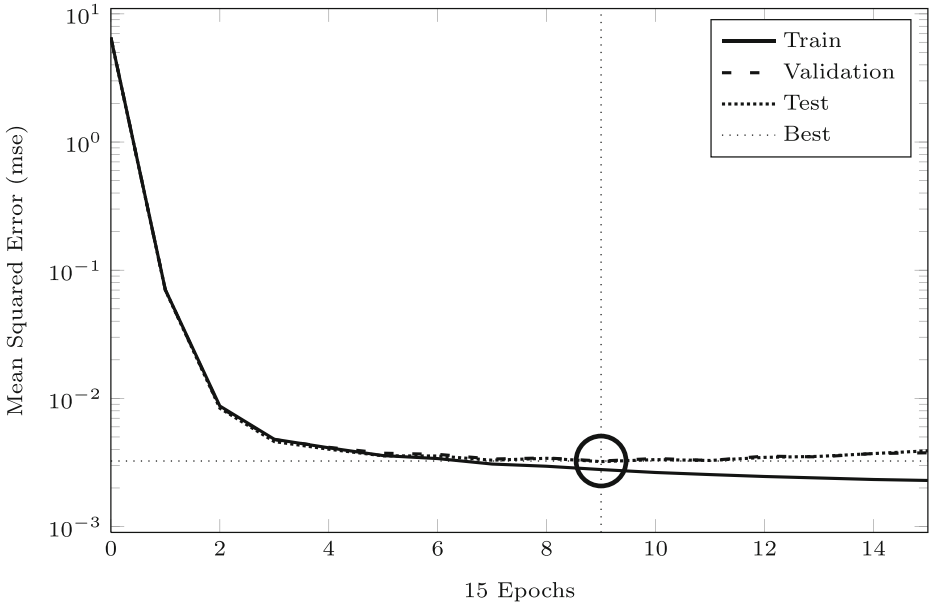

Fig. 4 Performance for stage 2, left eye. Best Validation Performance is 0.003246 at epoch 9

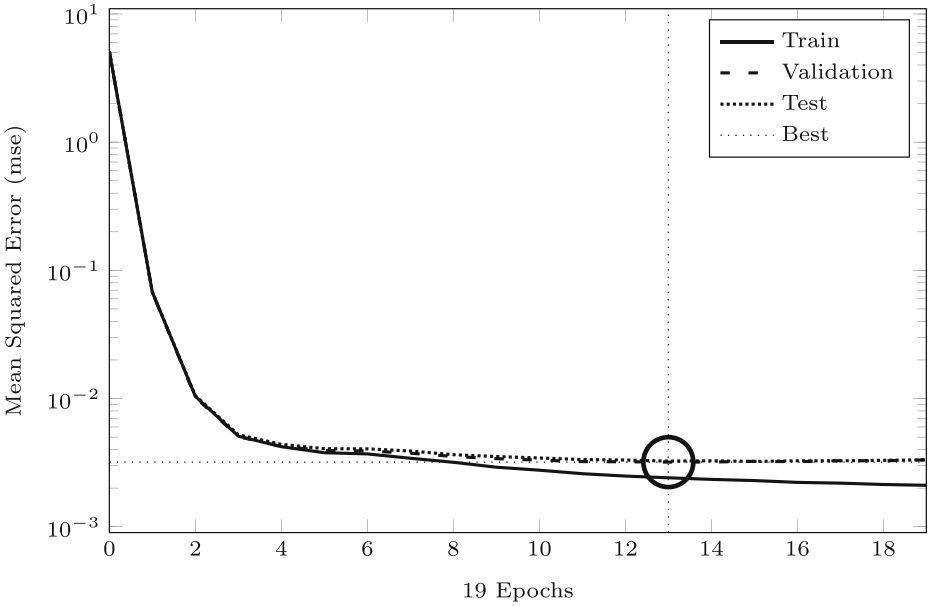

Fig. 5 Performance for stage 2, right eye. Best Validation Performance is 0.0031915 at epoch 13

A typical AdaBoost cascade for eyes from OpenCV contains approx. 100 weak classifiers. Thus the search operation requires about 90,000 floating point operations without feature extraction.

In the neural network regression the complexity is dominated by the largest layer. In our case the first layer of size 210 is fully connected to 20 neurons. Thus, a single evaluation requires approx. 5000 floating point operations and one relatively fast nonlinear function evaluation for each neuron. Detailed analysis of a cascade of two networks, presented later in the paper, shows that a single evauation requires approx. 16,000 floating point operations. When it comes to feature extraction, DCT has a complexity as FFT i.e $O(n \log(n))$ which is at order of $\sim 40,000$ for image of size 64×64 pixels. As such the proposed approach requires fewer floating point operations when compared to the state-of-the-art sliding window classifier.

4.3 Model validation

The eye localization algorithm is generally required to work on images obtained from the face detector, for example, cascade-based face detectors such as the one implemented in popular computer vision library OpenCV. In order to make the validation images as similar as possible to the real input images, face regions were selected by the cascade classifier *haarcascade_frontalface_alt.xml* with parameters min size (50,50), scale 1.1 and min neighbors 2. If the cascade found no face or more than one, such an example was removed.

4.3.1 BioID database and one-stage neural network

After the procedure of face detection, 1430 images were returned from the BioID database [1]. All the detected faces were randomly divided into two sets. The first 451

images were used for validation only, while the remaining 979 served as templates for training samples.

The validation set (as well as additional samples generated using this set) was not used for training; its only purpose was to measure the accuracy of eye localization. The accuracy can be described quantitatively in many different ways. The simplest measure is the mean squared error given by the neural network training tool. Unfortunately, this measure is not normalized. Therefore, we used the normalized eye localization error proposed in [10], which is defined in terms of the eye centre positions,

$$d_{eye} = \frac{\max(d_l, d_r)}{\|C_l - C_r\|}, \tag{4}$$

where C_l and C_r are the ground-truth positions and d_l and d_r are the Euclidean distances between the detected eye centres (left and right respectively) and their ground-truth positions. The accuracy of the eye localization algorithm is measured by the fraction of training samples in which the normalized error is lower than 0.1 or 0.25. The histogram of normalized error for neural network regression is shown in Fig. 6. Its empirical CDF is shown in Fig. 7. It should be noted that the most probable value of normalized relative error is ~ 0.04. We are dealing with images of $64 \times 64 px$ in which the eyes are usually more or less $25 px$ apart, thus the reported value of the normalized error corresponds to $\sim$ 1 px. We did not observe any examples with normalized error greater than 0.14.

The final results and comparison to other eye localization methods are collected in Table 1. The results described as 'Neural network regression' were derived from 451 unmodified testing images that were not used to generate additional samples. 'Neural network regression (full)' are the results obtained using all the 1430 images from the BioID database. None of those images was used during the training phase; however, a high number(979) were used as reference images to generate artificial samples.

4.3.2 Merged dataset and two-stage neural network

Although the BioID database is popular for validating algorithms, the images are far from being representative sample of faces. The images are of reasonable resolution and taken

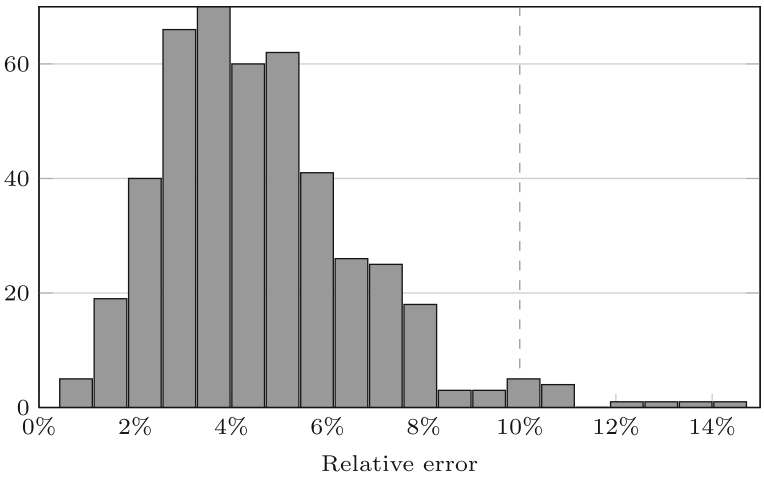

Fig. 6 Histogram of the relative error

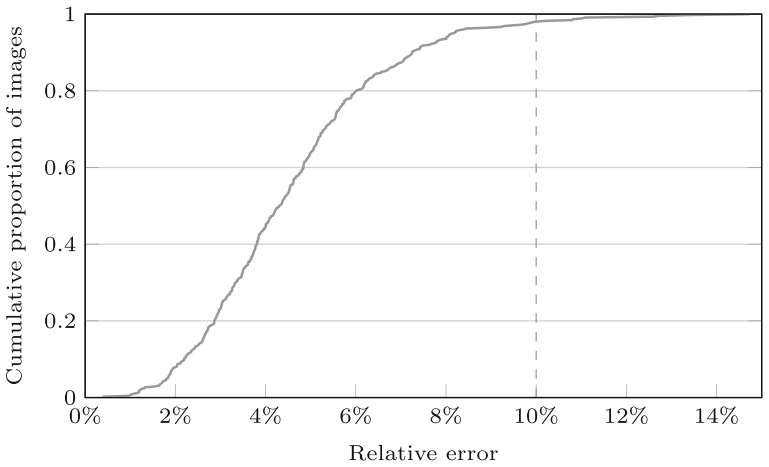

Fig. 7 Empirical cumulative distribution function of the relative error

under good light conditions. Moreover the database contains images of just a few adult subjects. The first observation is, that the networks trained on BioID performed poorly on the FG-NET [6] database and vice-versa. This was not a surprise, since the two are significantly different (FG-NET has a greater variety in age and image quality).

In order to test the proposed solution in a more realistic case, the BioID database was merged with the FG-NET database. The resulting collection of 2359 face images is referred to as *Set2*. Set2 was divided into two random subsets: training and testing. The training set consisted of 1652 images, with the remainder used for validation. Using the transformations described in Section 3, we obtained $\sim 100,000$ training samples and $\sim 14,000$ test samples. Note that no faces from the training appeared in the test set.

As it was already mentioned in Section 3.3, simple (single stage) neural network proved to be insufficient for our merged dataset. The largest network that can be train on our hardware had four hidden layers and twenty neurons in each layer (configuration 20-20-20-20). Even such a large network cannot achieve a satisfactory performance on Set2 (see Table 2).

However, the network performs well course-grained eye localization. In fact, more than 99 % predictions have an error lower than 7 pixels, which is approx. 25% of the average

Table 1 Eye localization accuracy for the BioID database

Method	d_{eye}	per-centage
HPF [21]	0.25	94.81 %
Isophote curvature [17]	0.1	90.9 %
scale Eyes + mouse [3]	0.1	93.2 %
Multiscale sparse dictionaries [20]	0.1	95.5 %
Multi-scale LBP [11]	0.1	97.9 %
Neural network regression	**0.1**	**98 %**
Neural network regression	**0.25**	**100 %**
Neural network regression (full)	**0.1**	**98.3 %**
Neural network regression (full)	**0.25**	**99.9 %**

Table 2 Eye localisation accuracy for BioID and Fg-Net

Method	d_{eye}	per-centage
Neural network regression stage 1	0.1	84.9 %
Neural network regression stage 1	0.25	99.3 %
Neural network regression stage $1+2$	**0.1**	**95.3 %**
Neural network regression stage $1+2$	**0.25**	**99.2 %**
SVR	0.25	99.5 %
SVR	0.1	89.8 %

eye distance which corresponds to the size of an eye. This is not wholly unexpected, considering that the features are low frequency DCT coefficients. Such features lack the detailed information required to locate the pupils. Instead they operate on the shape of the head and major facial features. In order to increase the accuracy, more coefficients can be used; however, this requires a larger training set to avoid overfitting, since the number of parameters in the neural network increases almost proportionally to the dimension of the feature vector.

Further, we prepared training and testing datasets as follows. First we took Set2 and split it into two subsets as previously. We used the training set (1652 images) to generate $\sim 100,000$ training samples. We obtained them by transforming the original 1652 images as described in Section 3. Hence, these samples were different from those used to train the first stage of the network. Then, we estimated the eye location using the first-stage network. Finally, we cut patches of $15 \times 15 px$ around the locations indicated by the first-stage neural network. These patches were later used to train the second-stage neural network. We generated another 14,000 samples transforming our testing set (707 images). We found approximate eyes locations in those samples using the first-stage neural network. We then cut $15 \times 15 px$ patches around those locations. The results from the second stage neural network obtained on those patches are reported as 2-stage regression.

This approach greatly increases accuracy (see Table 2 and Fig. 8). Additionally we performed tests using support vector regression. We used the same training and testing data sets as those used during the first stage of the two-stage regression model. The results obtained

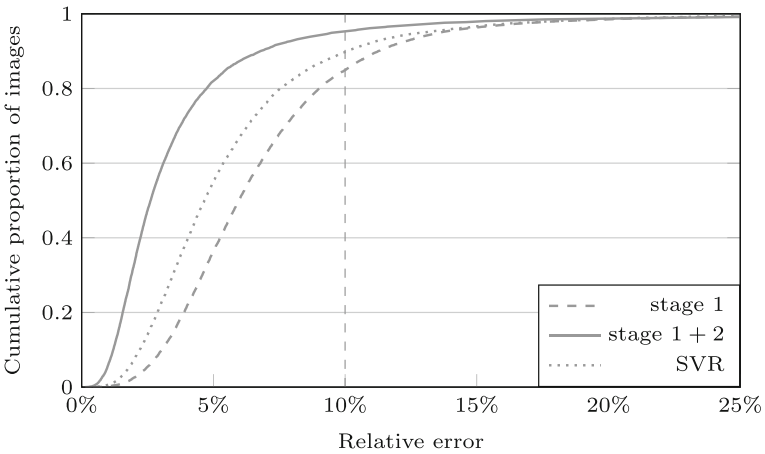

Fig. 8 Empirical cumulative distribution function of the relative error for Set2

in this experiment were better than those for a single stage neural network, although they were significantly worse than those obtained by the two-stage regression model. Finally, we can conclude that two-stage regression is significantly more accurate than a single network or Support Vector Regression.

5 Conclusion

Artificial feed-forward neural networks can be used for accurate eye region localization in face images. Moreover, the accuracy of a single network can be increased by combining outputs from two combined networks.

In contrast with typical classification models, this paper proposes the regression model. Such a model gives the final eye coordinates in a single evaluation step. It does not require extensive searching using the sliding window mechanism.

Using discrete cosine transform coefficients as the features makes it possible to reduce the network size while retaining most of the information stored in the image. Introducing a second network correcting the results of the first by using information from small image patch substantially improves the overall accuracy without significant complication or speed degradation.

These properties, combined with powerful procedure for generating artificial training samples makes it possible to train an accurate and efficient regression model. The accuracy of the proposed model was evaluated on a standard BioID database, and custom set was based on BioID and FG-NET. The results are comparable with state-of-the-art techniques. Since neural networks can be implemented efficiently, the proposed solution may be used in large-scale multimedia systems such as the OASIS Archive [4, 7].

Acknowledgments This work has been performed in the framework of the EU Project INDECT (*Intelligent information system supporting observation, searching and detection for security of citizens in urban environment*)–grant agreement number: 218086. This research was supported in part by PL-Grid Infrastructure.

References

1. BioID face database. URL http://www.bioid.com
2. Bishop C (2006) Pattern recognition and machine learning. Information science and statistics. Springer
3. Campadelli P, Lanzarotti R, Lipori G (2009) Precise eye and mouth localization. Int J Pattern Recognit. Artif Intell 23(03):359–377. doi:10.1142/S0218001409007259
4. Enge J, Głowacz A, Grega M, Leszczuk M, Papir Z, Romaniak P, Simko V (2009) Oasis archive – open archiving system with internet sharing. In: Mauthe A, Zeadally S, Cerqueira E, Curado M (eds) Future Multimedia Networking, *Lecture Notes in Computer Science*, vol. 5630, pp. 254–259. Springer, Berlin Heidelberg. doi:10.1007/978-3-642-02472-6_28
5. Everingham M, Zisserman A (2006) Regression and classification approaches to eye localization in face images. In: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, pp. 441–448
6. FG-NET Aging Databse. URL http://www.fgnet.rsunit.com. The link is not available at time of writing (since February 2013)
7. Głowacz A, Grega M, Leszczuk M, Papir Z, Romaniak P, Fornalski P, Lutwin M, Enge J, Lurk T, Šimko V Open internet gateways to archives of media art. Multimedia Tools Appl
8. Hudson Beale M., Hagan MT, Demuth HB (2012) Neural network toolbox user's guide. MathWorks, Natick
9. Ioannou S, Kessous L, Caridakis G, Karpouzis K, Aharonson V, Kollias S (2006) Adaptive on-line neural network retraining for real life multimodal emotion recognition. In: Kollias S, Stafylopatis A, Duch W,

Oja E (eds) Artificial Neural Networks ICANN 2006, *Lecture Notes in Computer Science*, vol. 4131, pp. 81–92. Springer, Berlin Heidelberg. doi:10.1007/11840817_9
10. Jesorsky O, Kirchberg KJ, Frischholz R (2001) Robust face detection using the hausdorff distance. In: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA '01, 90–95. Springer-Verlag, London
11. Kroon B, Maas S, Boughorbel S, Hanjalic A (2009) Eye localization in low and standard definition content with application to face matching. Comput Vis Image Underst 113(8):921–933. doi:10.1016/j.cviu.2009.03.013
12. Riopka T, Boult T (2003) The eyes have it. In: Proceedings of ACM SIGMM Multimedia Biometrics Methods and Applications Workshop, pp. 9–16
13. Rusek K, Guzik P (2013) Neural network regression of eyes location in face images. In: Dziech A, Czyżewski A (eds) Multimedia Communications, Services and Security, *Communications in Computer and Information Science*, vol. 368, pp. 204–212. Springer, Berlin Heidelberg. doi:10.1007/978-3-642-38559-9_18
14. Rusek K, Orzechowski T, Dziech A (2011) Lda for face profile detection. In: Dziech A, Czyżewski A (eds) Multimedia Communications, Services and Security, *Communications in Computer and Information Science*, vol. 149, pp. 144–148. Springer, Berlin Heidelberg. doi:10.1007/978-3-642-21512-4_17
15. Senechal T, Prevost L, Hanif S (2010). In: Schwenker F, Gayar N (eds) Artificial Neural Networks in Pattern Recognition, *Lecture Notes in Computer Science*, vol. 5998, pp. 141–148. doi:10.1007/978-3-642-12159-3_13. Springer, Berlin Heidelberg
16. Sun Y, Wang X, Tang X (2013) Deep convolutional network cascade for facial point detection. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 3476–3483. doi:10.1109/CVPR.2013.446
17. Valenti R, Gevers T (2008) Accurate eye center location and tracking using isophote curvature. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. doi:10.1109/CVPR.2008.4587529
18. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, pp. I–511–I–518 vol. 1. doi:10.1109/CVPR.2001.990517
19. Wilson PI, Fernandez J (2006) Facial feature detection using haar classifiers. J Comput Sci Coll 21(4):127–133
20. Yang F, Huang J, Yang P, Metaxas D (2011) Eye localization through multiscale sparse dictionaries. In: Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp. 514 –518. doi:10.1109/FG.2011.5771450
21. Zhou ZH, Geng X (2004) Projection functions for eye detection. Pattern Recognit 37(5):1049–1056. doi:10.1016/j.patcog.2003.09.006

Krzysztof Rusek is a PhD candidate at the Department of Telecommunications (AGH University of Science and Technology). He received M.Sc. in Electronics and Telecommunications in 2009 from the AGH University of Science and Technology.

His main interest is performance evaluation of telecommunications systems and queuing theory. He is also interest in computer vision and machine learning. He has actively participated in several 7th FP European program and national scientific projects (COST Action IC0703, INDECT, INSIGMA, TAPAS).

Piotr Guzik is a Ph.D. student at the Department of Telecommunications of AGH University of Science and Technology. He received his M.Sc. degree in astronomy from Jagiellonian University in 2009 (with honors). He has also received M.Sc. degree in applied computer science form AGH University of Science and Technology.

His research interests include image processing, digital watermarking, machine learning and computer vision. Since 2010 he had actively participated in both national, and international research projects, e.g., INDECT, INSIGMA, TAPAS, MAYDAY EURO 2012.